UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/623,023	07/18/2003	Robert J. Madril JR.	200208758-1	4467

22879 7590 05/15/2007
HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
DICKERSON, CHAD S	
ART UNIT	PAPER NUMBER
2625	
MAIL DATE	DELIVERY MODE
05/15/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.	**Applicant(s)**
10/623,023	MADRIL ET AL.
Examiner	**Art Unit**
Chad Dickerson	2625

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _18 July 2003_.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-36_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-36_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _7/18/2003_ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 112

1. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 8, 10, 16 and 24 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Re claim 8: the claim recites the limitation "the group" in line 12. There is insufficient antecedent basis for this limitation in the claim. It is suggested that the phrase “the group” should be changed to the phrase -- a group --.

Re claim 10: the claim recites the limitation "the group" in line 22. There is insufficient antecedent basis for this limitation in the claim. It is suggested that the phrase “the group” should be changed to the phrase -- a group --.

- the claim recites the limitation "the printer driver" in line 2 on page 3. There is insufficient antecedent basis for this limitation in the claim. It is suggested that the phrase “the printer driver” should be changed to -- a printer driver --.

Re claim 16: the claim recites the limitation "the group" in line 12. There is insufficient antecedent basis for this limitation in the claim. It is suggested that the phrase “the group” should be changed to the phrase -- a group --.

Re claim 24: the claim recites the limitation "the group" in line 13. There is insufficient antecedent basis for this limitation in the claim. It is suggested that the phrase "the group" should be changed to the phrase -- a group --.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1-6, 8-11, 13, 14, 19-22, 27-34 and 36 are rejected under 35 U.S.C. 102(e) as being anticipated by Kuno et al (US Pub No 2003/0135549).

Re claim 1: Kuno et al discloses print system including application server comprising processor-executable instructions configured for:

receiving a printer identification (i.e. the user edits information related to the desired document to be outputted and it informs the server of the printer type that will be used for local printing. The printer type can be considered as the printer identification; see fig. 6; paragraphs [0036] and [0037]);

generating a content type that identifies output that is compatible with a printer identified by the printer identification (i.e. the server then makes PDL data using the user's print request that is suitable for the registered printer designated by the user when informing the server (102) about the printer type. The content type that identifies or represents an output that is compatible to the printer is the PDL data that is generated by the server (102); see fig. 1, 5 and 6; paragraphs [0036] and [0037]); and

returning the content type (i.e. when the server returns the PDL data to the client PC to execute printing on the local printer, this performs the feature of returning the content type related to the output that is compatible to the printer identified; see fig. 1, 6; paragraphs [0036] and [0037]).

Re claim 2: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a processor-readable medium, further comprising processor-executable instructions configured for:

determining a printer driver that corresponds with the printer identification (i.e. when the user registers a printer that he or she wants to use and then informs the server of the printer type, this is an example of informing the server of the printer identification. Then the server (102) attempts to designate the printer name to locators of printer drivers and determine if there is a printer driver that corresponds to the printer type, or printer identification. Then the server installs the printer driver suitable to the designated printer; see figs. 1, 5 and 6; paragraphs [0036] and [0037]); and including in the content type, information identifying the printer driver (i.e. the information relating to the printer type is used to locate the printer drivers suitable for the printers identified. Therefore, this information is considered to identify printer drivers in the system; see figs. 1, 5 and 6; paragraphs [0036] and [0037]).

Re claim 3: The teachings of Kuno et al are disclosed above.

Kuno et al a processor-readable medium, further comprising processor-executable instructions configured for:

determining if the printer driver has been installed (i.e. in the system, the invention has the ability to allow the user to obtain a printer driver if it is not currently on the server. In order to know that a driver needs to be installed on the server, the determination that a printer driver is or is not on the server has to be made. Therefore, it is clear that the above feature is performed; see fig. 5 and 6; paragraphs [0036] and [0037]); and

automatically installing the printer driver if the printer driver has not already been installed (i.e. whenever a print request is received by the server (102), an automatic installation of a printer driver is performed. When the server realizes that the it does not have the appropriate printer driver to perform a print job, it can obtain a printer driver from a URL or some other source and install the printer driver; see fig. 1 and 5; paragraphs [0029] and [0036]).

Re claim 4: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a processor-readable medium, wherein the determining comprises:

accessing a lookup table (i.e. the server (102) has a table that relates the printer names to the locators of the printer drivers. This table is accessed by the server in order to acquire a suitable printer driver to the printer type; see fig. 5 and 6; paragraphs [0036] and [0037]); and

comparing the printer identification with printer driver entries in the lookup table (i.e. the printer names or types are related, or compared, to the locators of the printer drivers. Then the server installs the printer driver that is most suitable to the designated printer; see fig. 5 and 6; paragraphs [0036] and [0037]).

Re claim 5: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a processor-readable medium, further comprising processor-executable instructions configured for:

receiving a request to map a printer icon name to the content type (i.e. when the user requests a print, the server receives the request to link the printer icon name, that represents a printer driver, to the content type of the printer designated for printing, or printer type. The system clearly performs the above feature; see figs. 5 and 6; paragraphs [0036] and [0037]);

mapping a printer icon name based on the content type (i.e. the printer name is related, or mapped, to the printer driver that will output a specific content type to the local printer on the client PC. Therefore, the printer name, which represents a printer driver, is mapped or associated with a content type that will be outputted to the local printer; see figs. 5 and 6; paragraphs [0036] and [0037]); and

returning the printer icon name (i.e. when it is determined that the server does not have a certain printer driver, it can search for the printer driver by a URL. The URL will return the appropriate printer driver, or some data like a name that represents a printer driver, to the server (102) in order to process a certain content type; see figs. 1, 5 and 6; paragraphs [0029], [0036] and [0037]).

Re claim 6: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a processor-readable medium, wherein the printer icon name identifies a printer icon in a printer folder and the printer icon in the printer folder corresponds with the printer driver (i.e. the hard disk drive (116) stores printer drivers in the server (102). With the table of printer names and printer drivers being access by the server, it is clear that the locators that represent the printer drivers located on the hard disk drive (116) correspond to the printer drivers; see fig. 1, 5 and 6; paragraphs [0031], [0036] and [0037]).

Re claim 8: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a processor-readable medium, wherein the content type comprises information selected from the group comprising:

information identifying a printer driver (i.e. the server has information relating to the content type that identifies the printer driver stored in the hard disk drive (116); see fig. 2, 5 and 6; paragraphs [0031], [0033] and [0036]-[0039]);

information identifying capabilities of a printer driver (i.e. with the server (102) having the different drivers stored on the hard disk drive (116) related to the different client devices, this relates to and identifies information about the capability of the printer drivers present on the server; see fig. 2, 5 and 6; paragraphs [0031], [0033] and [0036]-[0039]);

information describing output that is compatible with the printer (i.e. when the server is looking for a suitable printer driver for the printer name, it looks at the table in the server and activates a driver. This driver activation is present on the monitor (118) on the server (102). Then the information that is outputted to the client PC is information that describes an output that is compatible with the printer; see fig. 2, 5 and 6; paragraphs [0031], [0033] and [0036]-[0039]);

information describing a compatible printer page description language (i.e. the print data outputted is used to be described with a page description language that is appropriate to the printer in which the print image data is outputted to; see fig. 2, 5 and 6; paragraphs [0031]-[0033] and [0036]-[0039]); and

information identifying a compatible printer (i.e. when the user sends to the server information designating a printer type that is used by the server to apply a certain printer driver, then information identifying a compatible printer is received from the client PC to the server; see fig. 2, 5 and 6; paragraphs [0031]-[0033] and [0036]-[0039]).

Re claim 9: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a processor-readable medium, wherein the receiving a printer identification includes receiving a request to map the printer identification to a content type (i.e. when the user requests a print, the server receives the request to link the printer icon name, that represents a printer driver, to the content type of the printer designated for printing, or printer type designated which is analogous to the printer identification. The system clearly performs the above feature; see figs. 5 and 6; paragraphs [0036] and [0037]).

Re claim 10: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a processor-readable medium wherein the printer identification is selected from the group comprising:

a text string specifying a printer model (i.e. the hard disk drive (116) stores the model names of printer that are used by the server to relate printer names to the printer driver; see fig. 2, 5 and 6; paragraphs [0031], [0036] and [0037]);

a text string specifying the printer driver (i.e. the hard disk drive (116) stores the names of printer drivers that are accessed by the server. The printer driver names specify the printer drivers in relation to the printer names; see fig. 2, 5 and 6; paragraphs [0031], [0036] and [0037]); and

a UNC (Universal Naming Convention) path (i.e. the registration information of client computers are stored in folders, or as directories, for access by the server. These folders or directories are analogous to a UNC since the directories represent a path to the registration information; see fig. 1 and 2; paragraphs [0031]) identifying a location of the printer driver (i.e. the system uses a URL, which is analogous to a UNC, to locate a printer driver that may be needed by the server; see fig. 2, 5 and 6; paragraphs [0029]-[0037]).

Re claim 11: Kuno et al discloses a print system including application server comprising processor-executable instructions configured for:

receiving a request to map a printer icon name to a content type (i.e. when the user requests a print, the server receives the request to link the printer icon name, that represents a printer driver, to the content type of the printer designated for printing, or printer type. The system clearly performs the above feature; see figs. 5 and 6; paragraphs [0036] and [0037]);

mapping a printer icon name to the content type (i.e. the printer name is related, or mapped, to the printer driver that will output a specific content type to the local printer on the client PC. Therefore, the printer name, which represents a printer driver, is mapped or associated with a content type that will be outputted to the local printer; see figs. 5 and 6; paragraphs [0036] and [0037]);

determining that a printer driver associated with the printer icon name is deficient (i.e. when the server makes a determination that the printer driver associated with, or related to, the printer icon name, or simply the printer name, is deficient, or the element is lacking in relation to the printer name, the server can use a URL or the client computer to obtain the suitable printer driver for the printer name. The determination is made that a printer driver suitable for a certain printer type may not be already installed in the server, due to other users requesting outputs using different printer types; see figs. 5 and 6; paragraphs [0029], [0030] and [0036]-[0045]);

searching for an alternate printer driver (i.e. when looking for another printer driver in accordance with another printer type, the server can search and find the other printer driver on the table with the printer names and printer driver locators. Searching for an alternate printer driver occurs when another printer of another content is using the server and the previous content type being used is not compatible with the current print request to print a document; see figs. 5 and 6; paragraphs [0029], [0030], [0036]-[0045]);

if an alternate printer driver is found, mapping an alternate printer icon name associated with the alternate printer driver (i.e. with another user requesting a document to be printed, the printer icon name, or simply the printer name, is a different name from the previous user. This is considered an alternate printer name. This alternate printer name is associated with the different printer driver used to process the content type compatible with the printer type given to the server; see figs. 5 and 6; paragraphs [0029], [0030], [0036]-[0045]); and

returning the alternate printer icon name (i.e. when it is determined that the server does not have a certain printer driver, it can search for the printer driver by a URL. The URL will return the appropriate printer driver, or some data like a name that represents a printer driver, to the server (102) in order to process a certain content type; see figs. 1, 5 and 6; paragraphs [0029], [0036] and [0037]).

Re claim 13: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a processor-readable medium, wherein the determining comprises:

accessing a lookup table (i.e. the server (102) has a table that relates the printer names to the locators of the printer drivers. This table is accessed by the server in order to acquire a suitable printer driver to the printer type; see fig. 5 and 6; paragraphs [0036] and [0037]); and

comparing the printer driver with entries in the lookup table (i.e. the printer names or types are related, or compared, to the locators of the printer drivers. Then the server installs the printer driver that is most suitable to the designated printer; see fig. 5 and 6; paragraphs [0036] and [0037]).

Re claim 14: Kuno et al discloses a print system application server comprising processor-executable instructions configured for:

receiving a request to map a printer icon name to a content type (i.e. when the user requests a print, the server receives the request to link the printer icon name, that represents a printer driver, to the content type of the printer designated for printing, or printer type. The system clearly performs the above feature; see figs. 5 and 6; paragraphs [0036] and [0037]);

mapping a printer icon name to the content type (i.e. the printer name is related, or mapped, to the printer driver that will output a specific content type to the local printer on the client PC. Therefore, the printer name, which represents a printer driver, is mapped or associated with a content type that will be outputted to the local printer; see figs. 5 and 6; paragraphs [0036] and [0037]);

determining if a printer driver associated with the printer icon name is available (i.e. when the user registers a printer that he or she wants to use and then informs the server of the printer type, this is an example of informing the server of the printer identification. Then the server (102) attempts to designate the printer name to locators of printer drivers and determine if there is a printer driver that corresponds to the printer type, or printer identification, is available to use. Then the server installs the printer driver suitable to the designated printer; see figs. 1, 5 and 6; paragraphs [0036] and [0037]); and

returning the printer icon name if the printer driver is available (i.e. when the CPU (110) checks the hard disk drive to see if a certain printer driver is available, the hard disk drive (116) returns to the CPU (110) the printer driver locator, analogous to a printer icon name, related to a certain printer. At this point, the printer driver may not necessarily need to be reinstalled. Also, when the server (102) checks a URL for a certain printer driver, the URL returns a printer driver in relation to the URL being searched; see figs. 1, 5 and 6; paragraphs [0036] and [0037]).

Re claim 19: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a processor-readable medium, further comprising processor-executable instructions configured for:

determining if the printer driver has been installed (i.e. in the system, the invention has the ability to allow the user to obtain a printer driver if it is not currently on the server. In order to know that a driver needs to be installed on the server, the determination that a printer driver is or is not on the server has to be made. Therefore, it is clear that the above feature is performed; see fig. 5 and 6; paragraphs [0036] and [0037]); and

automatically installing the printer driver if the printer driver has not already been installed (i.e. whenever a print request is received by the server (102), an automatic installation of a printer driver is performed. When the server realizes that the it does not have the appropriate printer driver to perform a print job, it can obtain a printer driver from a URL or some other source and install the printer driver; see fig. 1 and 5; paragraphs [0029] and [0036]).

Re claim 20: Kuno et al discloses a print system including application server comprising:

receiving a printer identification (i.e. the user edits information related to the desired document to be outputted and it informs the server of the printer type that will be used for local printing. The printer type can be considered as the printer identification; see fig. 6; paragraphs [0036] and [0037]);

generating a content type that identifies output compatible with a printer identified by the printer identification (i.e. the server then makes PDL data using the user's print request that is suitable for the registered printer designated by the user when informing the server (102) about the printer type. The content type that identifies or represents an output that is compatible to the printer is the PDL data that is generated by the server (102); see fig. 1, 5 and 6; paragraphs [0036] and [0037]); and

returning the content type (i.e. when the server returns the PDL data to the client PC to execute printing on the local printer, this performs the feature of returning the content type related to the output that is compatible to the printer identified; see fig. 1, 6; paragraphs [0036] and [0037]).

Re claim 21: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a method, further comprising:

determining a printer driver that corresponds with the printer identification (i.e. when the user registers a printer that he or she wants to use and then informs the server of the printer type, this is an example of informing the server of the printer identification. Then the server (102) attempts to designate the printer name to locators of printer drivers and determine if there is a printer driver that corresponds to the printer type, or printer identification. Then the server installs the printer driver suitable to the designated printer; see figs. 1, 5 and 6; paragraphs [0036] and [0037]); and including in the content type, information identifying the printer driver (i.e. the information relating to the printer type is used to locate the printer drivers suitable for the printers identified. Therefore, this information is considered to identify printer drivers in the system; see figs. 1, 5 and 6; paragraphs [0036] and [0037]).

Re claim 22: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a method, further comprising:

determining if the printer driver has been installed (i.e. in the system, the invention has the ability to allow the user to obtain a printer driver if it is not currently on the server. In order to know that a driver needs to be installed on the server, the determination that a printer driver is or is not on the server has to be made. Therefore, it is clear that the above feature is performed; see fig. 5 and 6; paragraphs [0036] and [0037]); and

automatically installing the printer driver if the printer driver has not already been installed (i.e. whenever a print request is received by the server (102), an automatic installation of a printer driver is performed. When the server realizes that the it does not have the appropriate printer driver to perform a print job, it can obtain a printer driver from a URL or some other source and install the printer driver; see fig. 1 and 5; paragraphs [0029] and [0036]).

Re claim 27: Kuno et al discloses a print system including application server comprising:

a content transformation harness configured to receive a document and a printer identification (i.e. the user edits information related to the desired document to be outputted and it informs the server of the printer type that will be used for local printing. The printer type can be considered as the printer identification. Also, the client PC can also send print data, which is considered to be a document, can be sent to the server to be processed into PDL data. Although a content transformation harness is not specifically disclosed, the feature is still performed; see figs. 5 and 6; paragraphs [0022]-[0025], [0031]-[0036] and [0037]);

a driver management service (102) configured to determine a printer driver corresponding with the printer identification (i.e. when the user registers a printer that he or she wants to use and then informs the server of the printer type, this is an example of informing the server of the printer identification. Then the server (102) attempts to designate the printer name to locators of printer drivers and determine if there is a printer driver that corresponds to the printer type, or printer identification. Then the server installs the printer driver suitable to the designated printer; see figs. 1, 5 and 6; paragraphs [0036] and [0037]) and to install the printer driver if the printer driver is not already installed (i.e. whenever a print request is received by the server (102), an automatic installation of a printer driver is performed. When the server realizes that the it does not have the appropriate printer driver to perform a print job, it can obtain a printer driver from a URL or some other source and install the printer driver; see fig. 1 and 5; paragraphs [0029] and [0036]); and a translator engine configured to transform the document into print-ready data using the printer driver (i.e. the printer drivers on the hard disk drive (116) are able to transform the print data into PDL or print-ready data; see fig. 5 and 6; paragraphs [0031]-[0037]).

Re claim 28: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a server, further comprising:

a content type, generated by the driver management service and containing information indicating the printer driver (i.e. the server (102) generates a content type that contains information that indicates the printer driver on the server (102) ; see fig. 2, 5 and 6; paragraphs [0031]-[0037]) and capabilities of the printer driver (i.e. with the hard disk drive (116) storing different printer drivers that are in relation to different client PCs using the system, this is information that shows the different capabilities that the printer drivers can perform on the server; see fig. 2, 5 and 6; paragraphs [0031]-[0037]).

Re claim 29: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a server, further comprising:

a driver look-up table containing printer driver entries and associated printer identifications (i.e. the server (102) has a table that relates the printer names to the locators of the printer drivers; see fig. 5 and 6; paragraphs [0036] and [0037]), the driver look-up table used by the driver management service to determine a printer driver (i.e.

this table is accessed by the server in order to acquire a suitable printer driver to the printer type; see fig. 5 and 6; paragraphs [0036] and [0037]).

Re claim 30: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a server, further comprising:

uninstalled driver files used by the driver management service to install the printer driver if the printer driver is not already installed (i.e. whenever a print request is received by the server (102), an automatic installation of a printer driver is performed. When the server realizes that it does not have the appropriate printer driver to perform a print job, it can obtain a printer driver from a URL or some other source and install the printer driver. The printer drivers that are present on the client device or the URL are considered to be uninstalled drivers since these drivers are not on the server. The server then acquires these drivers that are uninstalled and installs them on the server. This only occurs if this process is necessary; see fig. 1, 5 and 6; paragraphs [0029] and [0036]-[0045]).

Re claim 31: Kuno et al discloses a print system including application server comprising:

means for receiving a printer identification (i.e. the user edits information related to the desired document to be outputted and it informs the server of the printer type that will be used for local printing. The printer type can be considered as the printer identification; see fig. 6; paragraphs [0036] and [0037]);

means for generating a content type that identifies output compatible with a printer identified by the printer identification (i.e. the server then makes PDL data using the user's print request that is suitable for the registered printer designated by the user when informing the server (102) about the printer type. The content type that identifies or represents an output that is compatible to the printer is the PDL data that is generated by the server (102); see fig. 1, 5 and 6; paragraphs [0036] and [0037]); and

means for returning the content type (i.e. when the server returns the PDL data to the client PC to execute printing on the local printer, this performs the feature of returning the content type related to the output that is compatible to the printer identified; see fig. 1, 6; paragraphs [0036] and [0037]).

Re claim 32: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a server, further comprising:

means for determining a printer driver that corresponds with the printer identification (i.e. when the user registers a printer that he or she wants to use and then informs the server of the printer type, this is an example of informing the server of the printer identification. Then the server (102) attempts to designate the printer name to locators of printer drivers and determine if there is a printer driver that corresponds to the printer type, or printer identification. Then the server installs the printer driver suitable to the designated printer; see figs. 1, 5 and 6; paragraphs [0036] and [0037]); and means for including in the content type, information identifying the printer driver (i.e. the information relating to the printer type is used to locate the printer drivers suitable for the printers identified. Therefore, this information is considered to identify printer drivers in the system of the server (102); see figs. 1, 5 and 6; paragraphs [0036] and [0037]).

Re claim 33: The teachings of Kuno et al are disclosed above.

Kuno et al discloses a server, further comprising:

means for determining if the printer driver has been installed (i.e. in the system, the invention has the ability to allow the user to obtain a printer driver if it is not currently on the server. In order to know that a driver needs to be installed on the server, the determination that a printer driver is or is not on the server has to be made. Therefore, it is clear that the above feature is performed; see fig. 5 and 6; paragraphs [0036] and [0037]); and

means for automatically installing the printer driver if the printer driver has not already been installed (i.e. whenever a print request is received by the server (102), an automatic installation of a printer driver is performed. When the server realizes that the it does not have the appropriate printer driver to perform a print job, it can obtain a printer driver from a URL or some other source and install the printer driver; see fig. 1 and 5; paragraphs [0029] and [0036]).

Re claim 34: Kuno et al discloses a print system including application server comprising:

means for receiving a request to map a printer icon name to a content type (i.e. when the user requests a print, the server receives the request to link the printer icon name, that represents a printer driver, to the content type of the printer designated for printing, or printer type. The system clearly performs the above feature; see figs. 5 and 6; paragraphs [0036] and [0037]);

means for mapping a printer icon name to the content type (i.e. the printer name is related, or mapped, to the printer driver that will output a specific content type to the local printer on the client PC. Therefore, the printer name, which represents a printer driver, is mapped or associated with a content type that will be outputted to the local printer; see figs. 5 and 6; paragraphs [0036] and [0037]);

means for determining if a printer driver associated with the printer icon name is available (i.e. when the user registers a printer that he or she wants to use and then informs the server of the printer type, this is an example of informing the server of the printer identification. Then the server (102) attempts to designate the printer name to locators of printer drivers and determine if there is a printer driver that corresponds to the printer type, or printer identification, is available to use. Then the server installs the printer driver suitable to the designated printer; see figs. 1, 5 and 6; paragraphs [0036] and [0037]); and

means for returning the printer icon name if the printer driver is available (i.e. when the CPU (110) checks the hard disk drive to see if a certain printer driver is available, the hard disk drive (116) returns to the CPU (110) the printer driver locator, analogous to a printer icon name, related to a certain printer. At this point, the printer driver may not necessarily need to be reinstalled. Also, when the server (102) checks a URL for a certain printer driver, the URL returns a printer driver in relation to the URL being searched; see figs. 1, 5 and 6; paragraphs [0036] and [0037]).

Re claim 36: Kuno et al discloses a print system including application server comprising:

means for receiving a request to map a printer icon name to a content type (i.e. when the user requests a print, the server receives the request to link the printer icon name, that represents a printer driver, to the content type of the printer designated for printing, or printer type. The system clearly performs the above feature; see figs. 5 and 6; paragraphs [0036] and [0037]);

means for determining that a printer driver associated with a mapped printer icon name is deficient (i.e. when the server makes a determination that the printer driver associated with, or related to, the printer icon name, or simply the printer name, is deficient, or the element is lacking in relation to the printer name, the server can use a URL or the client computer to obtain the suitable printer driver for the printer name. The determination is made that a printer driver suitable for a certain printer type may not be already installed in the server, due to other users requesting outputs using different printer types; see figs. 5 and 6; paragraphs [0029], [0030] and [0036]-[0045]);

means for locating an alternate printer driver (i.e. when looking for another printer driver in accordance with another printer type, the server can search and find the other printer driver on the table with the printer names and printer driver locators. Searching for an alternate printer driver occurs when another printer of another content is using the server and the previous content type being used is not compatible with the current print request to print a document; see figs. 5 and 6; paragraphs [0029], [0030], [0036]-[0045]);

means for mapping an alternate printer icon name associated with the alternate printer driver (i.e. with another user requesting a document to be printed, the printer icon name, or simply the printer name, is a different name from the previous user. This is considered an alternate printer name. This alternate printer name is associated with the different printer driver used to process the content type compatible with the printer type given to the server; see figs. 5 and 6; paragraphs [0029], [0030], [0036]-[0045]); and

means for returning the alternate printer icon name (i.e. when it is determined that the server does not have a certain printer driver, it can search for the printer driver by a URL. The URL will return the appropriate printer driver, or some data like a name that represents a printer driver, to the server (102) in order to process a certain content type; see figs. 1, 5 and 6; paragraphs [0029], [0036] and [0037]).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kuno et al in view of Gunji (US Pub No 2003/0065755).

Re claim 7: The teachings of Kuno et al are disclosed above.

However, Kuno et al fails to teach a processor-readable medium, wherein the content type is a structured XML (Extensible Markup Language) document.

However, this is well known in the art as evidenced by Gunji. Gunji discloses the content type is a structured XML (Extensible Markup Language) document (i.e. the invention has a method where an XML type is used to describe attribute information that specifies the attributes of a driver provided on a server; see paragraphs [0012]-[0016]).

Therefore, in view of Gunji, it would have been obvious to one of ordinary skill at the time the invention was made to have a content type structured as an XML (Extensible Markup Language) document in order to analyze attribute information of a driver in a certain language (as stated in Gunji paragraph [0013]).

7. Claims 12, 15, 25 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kuno et al in view of Liang '297 (US Pat No 7212297).

Re claim 12: The teachings of Kuno et al are disclosed above.

Kuno et al teaches a processor-readable medium, further comprising processor-executable instructions configured for returning an alternate printer driver.

However, Kuno et al fails to teach returning a message indicating that the printer driver is not available if an alternate printer driver is not found.

However, this is well known in the art as evidenced by Liang '297. Liang '297 discloses returning a message indicating that the printer driver is not available if an alternate printer driver is not found (i.e. if a conventional printer driver is not available, an error message is generated. The system searches for printer drivers on the system and if there are no printer drivers on the system, both an original and an alternative printer driver are not found on the system; see fig. 8 and 9; col. 7, lines 4-59).

Therefore, in view of Liang '297, it would have been obvious to one of ordinary skill at the time the invention was made to return a message indicating that the printer driver is not available if an alternate printer driver is not found in order to generate an error message when a printer driver is not available (as stated in Liang '297 col. 7, lines 4-59).

Re claim 15: The teachings of Kuno et al are disclosed above.

However, Kuno et al fails to teach a processor-readable medium, further comprising processor-executable instructions configured for returning an error message if the printer driver is not available.

However, this is well known in the art as evidenced by Liang '297. Liang '297 discloses returning an error message if the printer driver is not available (i.e. if a conventional printer driver is not available, an error message is generated; see fig. 8 and 9; col. 7, lines 4-59).

Therefore, in view of Liang '297, it would have been obvious to one of ordinary skill at the time the invention was made to return an error message if the printer driver is not available incorporated in the device of Kuno et al in order to generate an error message when a printer driver is not available (as stated in Liang '297 col. 7, lines 4-59).

Re claim 25: The teachings of Kuno et al are disclosed above.

Kuno et al teaches a method comprising:

receiving a request to map a printer icon name to a content type (i.e. when the user requests a print, the server receives the request to link the printer icon name, that represents a printer driver, to the content type of the printer designated for printing, or printer type. The system clearly performs the above feature; see figs. 5 and 6; paragraphs [0036] and [0037]);

determining that a printer driver associated with the printer icon name is deficient (i.e. when the server makes a determination that the printer driver associated with, or related to, the printer icon name, or simply the printer name, is deficient, or the element is lacking in relation to the printer name, the server can use a URL or the client computer to obtain the suitable printer driver for the printer name. The determination is made that a printer driver suitable for a certain printer type may not be already installed in the server, due to other users requesting outputs using different printer types; see figs. 5 and 6; paragraphs [0029], [0030] and [0036]-[0045]); and

However, Kuno et al fails to teach returning a message indicating that the printer driver is not available.

However, this is well known in the art as evidenced by Liang '297. Liang '297 discloses returning a message indicating that the printer driver is not available (i.e. when the system checks for a conventional driver and it is not present, a error message is generated indicating that the printer driver is not available; see fig. 8 and 9; col. 7, lines 4-59).

Therefore, in view of Liang '297, it would have been obvious to one of ordinary skill at the time the invention was made to return a message indicating that the printer driver is not available in order to generate an error message when a printer driver is not available (as stated in Liang '297 col. 7, lines 4-59).

Re claim 26: The teachings of Kuno et al in view of Liang '297 are disclosed above. Kuno et al teaches a method, further comprising:

searching for an alternate printer driver (i.e. when looking for another printer driver in accordance with another printer type, the server can search and find the other printer driver on the table with the printer names and printer driver locators. Searching for an alternate printer driver occurs when another printer of another content is using the server and the previous content type being used is not compatible with the current print request to print a document; see figs. 5 and 6; paragraphs [0029], [0030], [0036]-[0045]);

if an alternate printer driver is found, mapping an alternate printer icon name associated with the alternate printer driver (i.e. with another user requesting a document to be printed, the printer icon name, or simply the printer name, is a different name from the previous user. This is considered an alternate printer name. This alternate printer name is associated with the different printer driver used to process the content type compatible with the printer type given to the server; see figs. 5 and 6; paragraphs [0029], [0030], [0036]-[0045]); and

returning the alternate printer icon name for the alternate printer driver (i.e. when it is determined that the server does not have a certain printer driver, it can search for the printer driver by a URL. The URL will return the appropriate printer driver, or some data like a name that represents a printer driver, to the server (102) in order to process a certain content type; see figs. 1, 5 and 6; paragraphs [0029], [0036] and [0037]).

8. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kuno et al, as modified by Liang '297, and further in view of Chang et al (US Pub No 2002/0059415).

Re claim 16: The teachings of Kuno et al in view of Liang '297 are disclosed above.

Liang '297 teaches a processor-readable medium, wherein the error message is selected from the group comprising:

a message indicating that the printer driver is unavailable (i.e. if a conventional printer driver is not available, an error message is generated after the determination of a printer driver is not available. This message indicates that a printer driver is not available; see fig. 8 and 9; col. 7, lines 4-59); and

a message indicating that the printer driver is temporarily unavailable (i.e. if a conventional printer driver is not available, an error message is generated after the determination of a printer driver is not available. This message indicates that a printer driver is not available. Whether temporarily unavailable or unavailable, under the same condition, the printer driver is unavailable; see fig. 8 and 9; col. 7, lines 4-59).

However, Kuno et al in view of Liang '297 fails to teach that the request should be attempted later.

However, this is well known in the art as evidenced by Chang et al. Chang et al discloses that the request should be attempted later (i.e. the system notifies the user to try again or it notifies that user at a later time for the use of the printer, through the printer driver, when the actual requested service is available; see fig. 5; paragraph [0148]).

Therefore, in view of Chang et al, it would have been obvious to one of ordinary skill at the time the invention was made to have a message indicating that the request should be attempted later in order to notify the user when services become available (as stated in Chang et al paragraph [0148]).

9. Claims 17, 18, 23 and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kuno et al in view of Terrill et al (US Pub No 2002/0188646).

Re claim 17: The teachings of Kuno et al are disclosed above.

Kuno et al teaches determining that the printer driver is available and determining that the printer driver is not available (i.e. in the system, the invention has the ability to allow the user to obtain a printer driver if it is not currently on the server. In order to know that a driver needs to be installed on the server, the determination that a printer driver is or is not on the server has to be made. Therefore, it is clear that the system determines if a printer driver is or is not available. If the printer driver is available, then it can be immediately used and installation of a printer driver is not necessary, which is stated in paragraph [0037]. If it is determined that the printer driver is not available, the server obtains a printer driver suitable for outputting the content type; see fig. 5 and 6; paragraphs [0036] and [0037]).

However, Kuno et al fails to teach comparing a threshold number of print requests to a current number of print requests, if the current number is less than the threshold number and if the threshold number does not exceed the current number.

However, this is well known in the art as evidenced by Terrill et al. Terrill et al discloses

comparing a threshold number of print requests to a current number of print requests (i.e. the system discloses the a threshold of a number of print jobs or print jobs completed. The system compares the pre or post print information and the print job completion data to constraints in the system. If constraints in the system are reached, data is transferred to a management server; see paragraphs [0011], [0038] and [0054]);

if the current number is less than the threshold number (i.e. when comparing the print job completion information, post and pre print information with the threshold, the information is compared to see if the threshold is reached. If the threshold is not reached, it represents the information being compared to the threshold is lower than the threshold; see paragraphs [0011], [0038] and [0054]); and

if the threshold number does not exceed the current number (i.e. when the print job information attains the threshold of the system, then information from the data store is transferred to the management server. This is an example of the threshold not exceeding the current number or the print job number, but being equal to that number; see paragraphs [0011], [0038] and [0054]).

Therefore, in view of Terrill et al, it would have been obvious to one of ordinary skill at the time the invention was made to compare a threshold number of print requests to a current number of print requests, to judge if the current number is less than the threshold number or if the threshold number does not exceed the current number in order to judge to realization of the threshold in the system (as stated in Terrill et al paragraph [0011]).

Re claim 18: The teachings of Kuno et al in view of Terrill et al are disclosed above.

Kuno et al teaches a processor-readable medium, wherein the comparing comprises:

accessing a look-up table (i.e. the server (102) has a table that relates the printer names to the locators of the printer drivers. This table is accessed by the server in order to acquire a suitable printer driver to the printer type; see fig. 5 and 6; paragraphs [0036] and [0037]); and

determining the current number according to active print requests being processed by the printer driver (i.e. on the server (102), the hard disk drive (116) stores the current number of active printer drivers corresponding to the print request given from the client. With this information, the server (102) is able to determine the current number of print jobs because of the print requests that are correlated to the active printer drivers in the server that is processing the print requests; see fig. 1, 2, 5 and 6; see paragraphs [0029]-[0033]).

However, Kuno et al fails to teach a processor-readable medium, wherein the comparing comprises: accessing the threshold number from a look-up table.

However, this is well known in the art as evidenced by Terrill et al. Terrill et al discloses the comparing comprises:

accessing the threshold number from a look-up table (i.e. in the conventional in Terrill et al, job information is listed in a job table. In the system, the thresholds are contained in the data transfer module (214), data store (212) or another location. Although the system does not specifically state accessing a threshold number from a look-up table, the threshold number is accessed from a various amounts of locations. If the threshold was on the job information collection and correlation module which has the job table, it may perform the feature of being accessed from a look-up table; see paragraphs [0004], [0037] and [0038]).

Therefore, in view of Terrill et al, it would have been obvious to one of ordinary skill at the time the invention was made to access the threshold number from a look-up table in order to obtain and store information in job table (as stated in Terrill et al paragraph [0004]).

Re claim 23: The teachings of Kuno et al are disclosed above.

Kuno et al teaches a method comprising:

receiving a request to map a printer icon name to a content type (i.e. when the user requests a print, the server receives the request to link the printer icon name, that represents a printer driver, to the content type of the printer designated for printing, or printer type. The system clearly performs the above feature; see figs. 5 and 6; paragraphs [0036] and [0037]);

mapping a printer icon name to the content type (i.e. the printer name is related, or mapped, to the printer driver that will output a specific content type to the local printer on the client PC. Therefore, the printer name, which represents a printer driver, is mapped or associated with a content type that will be outputted to the local printer; see figs. 5 and 6; paragraphs [0036] and [0037]);

associating a printer driver with the printer icon name (i.e. when it is determined that the server does not have a certain printer driver, it can search for the printer driver by a URL. The URL will return the appropriate printer driver, or some data like a name that represents a printer driver, to the server (102) in order to process a certain content type. The printer driver returned is associated with the printer name that is designated by the user; see figs. 1, 5 and 6; paragraphs [0029], [0036] and [0037]);

determining that the printer driver is available and determining that the printer driver is not available (i.e. in the system, the invention has the ability to allow the user to obtain a printer driver if it is not currently on the server. In order to know that a driver needs to be installed on the server, the determination that a printer driver is or is not on the server has to be made. Therefore, it is clear that the system determines if a printer driver is or is not available. If the printer driver is available, then it can be immediately used and installation of a printer driver is not necessary, which is stated in paragraph [0037]. If it is determined that the printer driver is not available, the server obtains a printer driver suitable for outputting the content type; see fig. 5 and 6; paragraphs [0036] and [0037]).

returning the printer icon name if the printer driver is available (i.e. when the CPU (110) checks the hard disk drive to see if a certain printer driver is available, the hard disk drive (116) returns to the CPU (110) the printer driver locator, analogous to a printer icon name, related to a certain printer. At this point, the printer driver may not necessarily need to be reinstalled. Also, when the server (102) checks a URL for a certain printer driver, the URL returns a printer driver in relation to the URL being searched; see figs. 1, 5 and 6; paragraphs [0036] and [0037]).

However, Kuno et al fails to teach comparing a threshold number of print requests to a current number of print requests, if the current number is less than the threshold number and if the threshold number does not exceed the current number.

However, this is well known in the art as evidenced by Terrill et al. Terrill et al discloses

comparing a threshold number of print requests to a current number of print requests (i.e. the system discloses the a threshold of a number of print jobs or print jobs completed. The system compares the pre or post print information and the print job completion data to constraints in the system. If constraints in the system are reached, data is transferred to a management server; see paragraphs [0011], [0038] and [0054]);

if the current number is less than the threshold number (i.e. when comparing the print job completion information, post and pre print information with the threshold, the information is compared to see if the threshold is reached. If the threshold is not reached, it represents the information being compared to the threshold is lower than the threshold; see paragraphs [0011], [0038] and [0054]); and

if the threshold number does not exceed the current number (i.e. when the print job information attains the threshold of the system, then information from the data store is transferred to the management server. This is an example of the threshold not exceeding the current number or the print job number, but being equal to that number; see paragraphs [0011], [0038] and [0054]).

Therefore, in view of Terrill et al, it would have been obvious to one of ordinary skill at the time the invention was made to compare a threshold number of print requests to a current number of print requests, to judge if the current number is less than the threshold number or if the threshold number does not exceed the current number in order to judge to realization of the threshold in the system (as stated in Terrill et al paragraph [0011]).

Re claim 35: The teachings of Kuno et al are disclosed above.

Kuno et al teaches a server, wherein the means for determining comprises:

means for determining that the printer driver is available and means for determining that the printer driver is not available (i.e. in the system, the invention has the ability to allow the user to obtain a printer driver if it is not currently on the server. In order to know that a driver needs to be installed on the server, the determination that a printer driver is or is not on the server has to be made. Therefore, it is clear that the system determines if a printer driver is or is not available. If the printer driver is available, then it can be immediately used and installation of a printer driver is not necessary, which is stated in paragraph [0037]. If it is determined that the printer driver is not available, the server obtains a printer driver suitable for outputting the content type; see fig. 5 and 6; paragraphs [0036] and [0037]).

However, Kuno et al fails to teach a means for comparing a threshold number of print requests to a current number of print requests, if the current number is less than the threshold number and if the threshold number does not exceed the current number.

However, this is well known in the art as evidenced by Terrill et al. Terrill et al discloses

means for comparing a threshold number of print requests to a current number of print requests (i.e. the system discloses the a threshold of a number of print jobs or print jobs completed. The system compares the pre or post print information and the print job completion data to constraints in the system. If constraints in the system are reached, data is transferred to a management server; see paragraphs [0011], [0038] and [0054]);

if the current number is less than the threshold number (i.e. when comparing the print job completion information, post and pre print information with the threshold, the information is compared to see if the threshold is reached. If the threshold is not reached, it represents the information being compared to the threshold is lower than the threshold; see paragraphs [0011], [0038] and [0054]); and

if the threshold number does not exceed the current number (i.e. when the print job information attains the threshold of the system, then information from the data store is transferred to the management server. This is an example of the threshold not exceeding the current number or the print job number, but being equal to that number; see paragraphs [0011], [0038] and [0054]).

Therefore, in view of Terrill et al, it would have been obvious to one of ordinary skill at the time the invention was made to have a means for comparing a threshold number of print requests to a current number of print requests, to judge if the current number is less than the threshold number or if the threshold number does not exceed the current number in order to judge to realization of the threshold in the system (as stated in Terrill et al paragraph [0011]).

10. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kuno et al, modified by Terrill et al and Liang '297, and further in view of Chang et al.

Re claim 24: The teachings of Kuno et al in view of Terrill et al are disclosed above.

However, Kuno et al in view of Terrill et al fails to teach a method, further comprising returning an error message if the printer driver is not available, the error message selected from the group comprising: a message indicating that the printer driver is unavailable; and a message indicating that the printer driver is temporarily unavailable and

Liang '297 teaches a processor-readable medium, wherein the error message is selected from the group comprising:

a message indicating that the printer driver is unavailable (i.e. if a conventional printer driver is not available, an error message is generated after the determination of a printer driver is not available. This message indicates that a printer driver is not available; see fig. 8 and 9; col. 7, lines 4-59); and a message indicating that the printer driver is temporarily unavailable (i.e. if a conventional printer driver is not available, an error message is generated after the determination of a printer driver is not available. This message indicates that a printer driver is not available. Whether temporarily unavailable or unavailable, under the same condition, the printer driver is unavailable; see fig. 8 and 9; col. 7, lines 4-59).

However, Kuno et al, modified by Terrill et al, and further in view of Liang '297 fails to teach that the request should be attempted later.

However, this is well known in the art as evidenced by Chang et al. Chang et al discloses that the request should be attempted later (i.e. the system notifies the user to try again or it notifies that user at a later time for the use of the printer, through the printer driver, when the actual requested service is available; see fig. 5; paragraph [0148]).

Therefore, in view of Chang et al, it would have been obvious to one of ordinary skill at the time the invention was made to have a message indicating that the request should be attempted later in order to notify the user when services become available (as stated in Chang et al paragraph [0148]).

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Fu et al (US Pat No 7064856) discloses a system that discloses receiving printer identification that identifies a certain content that can be interpreted and a generation of an output content type that is compatible with a client's printer and returns that output content type to the client. This feature is shown in figures 2 and 3.

Roosen (US Pub No 2003/0048473) discloses searching for an alternate printer driver and returning an alternate printer driver to perform the processing of a print job received from a computer.

Bergstrand (US Pat 7184159) discloses a system that discloses receiving printer identification that identifies a certain printer with a specific content, generating a specific content that is compatible to a certain printer and returning that content to a destination. This is illustrated in figures 4 and 5.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chad Dickerson whose telephone number is (571)-270-1351. The examiner can normally be reached on Mon. thru Thur. 9:00-6:30 Fri. 9:00-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Aung Moe can be reached on (571)- 272-7314. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

CD/
Chad Dickerson
May 11, 2007

AUNG S. MOE
SUPERVISORY PATENT EXAMINER
5/11/07